Adaptive Behavior

Re-embodiment of Honeybee Aggregation Behavior in an Artificial Micro-Robotic System

Serge Kernbach[1], Ronald Thenius[2], Olga Kernbach[1], Thomas Schmickl[2]
[1] *Institute of Parallel and Distributed Systems, University of Stuttgart, Germany*
[2] *Department of Zoology, University of Graz, Austria*

In this article we describe the re-embodiment of biological aggregation behavior of honeybees in Jasmine micro-robots. The observed insect behavior, in the context of the insect's sensor–actor system, is formalized as behavioral and motion-sensing meta-models. These meta-models are transformed into a sensor–actor system of micro-robots by means of a sensors virtualization technique. This allows us to keep the efficiency and scalability of the bio-inspired approach. We also demonstrate the systematic character of this re-embodiment procedure on collective aggregation in a real robotic swarm.

Keywords biologically inspired algorithms · micro-robotic swarm · aggregation behavior · embodiment · sensor–actor couplings

1 Introduction

Swarm intelligence is a phenomenon found in social insects (Bonabeau, Dorigo, & Theraulaz, 1999). This feature of distributed self-organized processes is characterized by the ability of a group of animals to solve a common problem collectively. Usually, a single worker animal is not capable of solving a problem alone, but solving it collectively produces an increase in the collective fitness of the colony. Examples of swarm intelligence include the selection of foraging targets performed by ants and honeybees; the self-organized nest construction by ants, bees, termites and wasps; and sorting tasks.

Biological strategies of collective problem solving can successfully be used in artificial micro-robotic systems (Sahin, 2004). Micro-robots cannot act individually in solving problems (I-Swarm, 2007), because of their limitations in size, available on-board energy, actuation, sensors, and communication capabilities. Technically useful behavior can be achieved not only when robots coordinate their behavior but also when a problem-solving strategy is designed for a collective solution (Kornienko, Kornienko, & Levi, 2004). Swarm-robotic experiments clearly demonstrated that increasing the number of robots that work on one problem in parallel, does not lead to linear or sub-linear improvement of collective performance (e.g., Jimenez, 2005). Moreover, the growth of a robot population over some threshold decreases collective fitness because of overhead and deadlock in communication and coordination. Simple parallel solutions do not increase the fitness of a robot swarm.

Biological solutions possess structures that are required by most problem-solving approaches (Camazine et al., 2003) because of their long-term natural optimization via evolution. The key issues are efficient and scalable coordination mechanisms, specific incorporation of global and local information sources as control signals, and full independence of individual

Correspondence to: S. Kernbach, Institute of Parallel and Distributed Systems, University of Stuttgart, Universitätsstr. 38, D-70569 Stuttgart, Germany. *E-mail*: korniesi@ipvs.uni-stuttgart.de.
Tel.: +49-711-7816373, *Fax*: +49-711-7816250

(2009), Vol 17(3): 237–259.
DOI: 10.1177/1059712309104966

behavior, where even a large number of agents do not hinder each other. Unfortunately, these highly optimized natural solutions cannot be directly applied to robotic systems. The point is that the individual behavior of each biological agent reflects its own sensor and actuator system. These systems are too different in insects and micro-robots to be directly transferred from one entity to the other. Indeed, we believe the "silicon–metal" robot cannot, even in the near future, mimic the sensor and actor capabilities of real biological organisms.

To use the advantages of biological strategies for collective robotic systems, these strategies have to be re-embodied for another sensor–actor system (Kornienko, Kornienko, & Levi, 2005a). Here we encounter the challenge of retaining the efficiency of biologically inspired approaches despite their different final implementation. In this work we demonstrate not only this particular re-embodiment procedure, but also its systematic nature by means of a virtualization technique. This is an important topic for the swarm-robotic and adaptive-behavior research community as an adaptation approach to biologically inspired mechanisms for real large-scale swarms.

The rest of the article is organized as follows. The experiments with honeybees thermotactic aggregation behavior is described in Section 2. Re-embodiment of bio-inspired strategies for robotic system is presented in Section 3, the experiments in Section 4 and our conclusions in Section 5.

2 Honeybees Thermotactic Aggregation Behavior

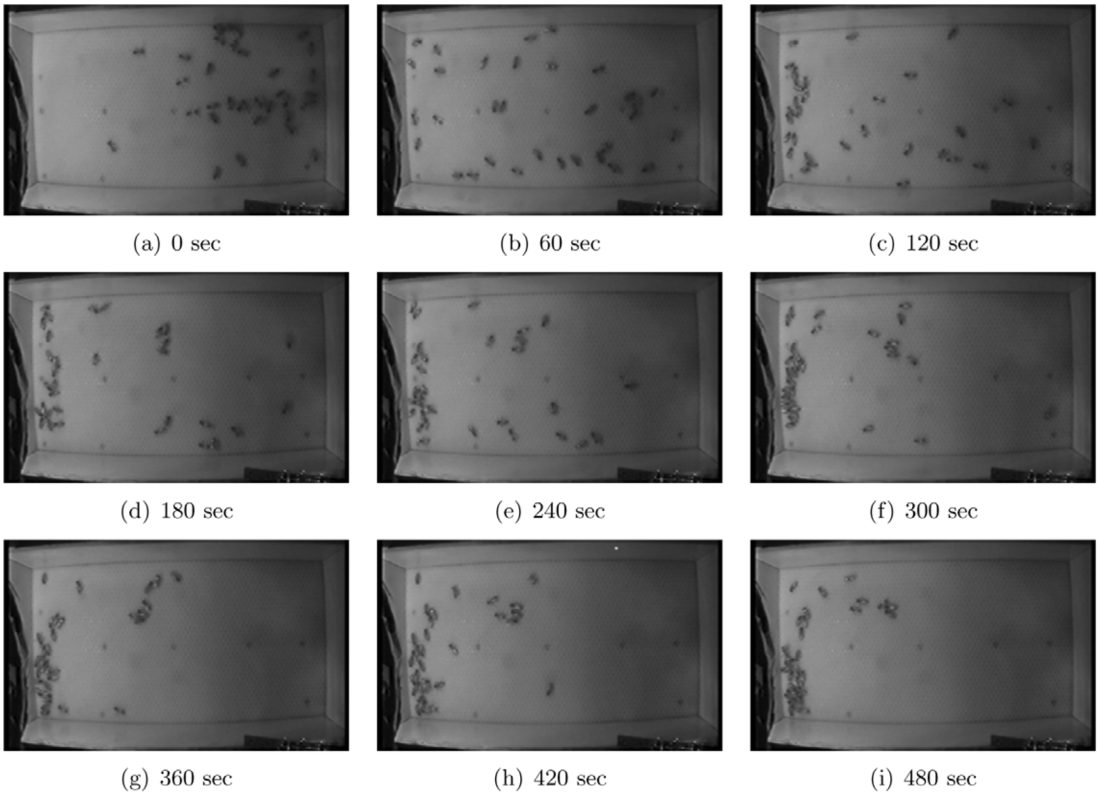
Thermotactic aggregation behavior found in honeybees (Crailsheim, Eggenreich, Ressi, & Szolderits, 1999; Grodzicki & Caputa, 2005; Heran, 1952) is used as the test scenario. On flat temperature gradients, complex collective behavior is observed. This behavior is expressed by the formation and disaggregation of clusters throughout the temperature field. After some time a single large cluster emerges, which is located at (or near) the region of optimal temperature (see Figure 1).

This collective behavior shows some analogies to other collective behaviors found in animals such as clustering and aggregating of animals via positive feedback, and "sorting" problems.

On the one hand is item-sorting behavior, found in ants, termites and bees (Theraulaz & Bonabeau, 1995). Items such as "food" or "brood" are transported by the workers from one place to another and aggregated in several distinct places. Often the workers aggregate items of different kinds in different places, thus they perform qualitative sorting. In this way, seeds can be sorted by size or by kind, or brood stages can be sorted, so that eggs, larvae, and pupae are separated. The process of sorting and aggregating items is purely self-organized, but often it is also modulated by environmental conditions (humidity, temperature, distance to queen etc.). If the aggregated items are the building blocks of the nest (sand, mud, etc.), then the same process is used for nest construction. This is found in the dynamic shaping of the queen's chamber in termites and in the radial nest structure of some ant species (Bonabeau et al., 1999; Camazine et al., 2003).

On the other hand, animals often aggregate themselves (animal sorting) in specific areas. The most prominent cases are the reproductive swarms and the winter clusters of honeybees (Myerscough, 1993; Sumpter & Broomhead, 2000; Watmough & Camazine, 1995). Other examples include: aggregation of ants in the nest (Theraulaz et al., 2002); chain formation in ants (Lioni, Sauwens, Theraulaz, & Deneubourg, 2001); and the aggregation of cockroaches (Jeanson et al., 2005; Jost et al., 2004), bark beetle (Camazine et al., 2003), slime-mold amoebas, and several bacteria species (Camazine et al., 2003). All these phenomena work via simple positive feedback loops and they all have in common that aggregation is performed without any central regulatory unit. These examples provide bio-inspiration for technical approaches to aggregation scenarios in real robot swarms.

In this article we propose a biologically inspired aggregation algorithm for mobile micro-robots based on thermotactic behavior found in honeybees. Aggregation of honeybees is strongly dependent on the form of the temperature gradient. The bees find the point of optimal (high) temperature, based on the difference between the warmest and the coldest point in the arena. If the gradient is not very steep, the bees are not able to determine the uphill direction in the gradient, and mainly perform a random walk. Nevertheless, they are still able to find the place of the optimal temperature, but they can only find it collectively.



(a) 0 sec (b) 60 sec (c) 120 sec

(d) 180 sec (e) 240 sec (f) 300 sec

(g) 360 sec (h) 420 sec (i) 480 sec

Figure 1 A group of 32 bees aggregating on a flat gradient. The bees were released in the colder right part of the arena (approx. 25 °C). After some time, the bees aggregate on the left side of the arena, which was 36 °C. During the runs, bees formed clusters in colder areas of the arena, but most of the bees moved to the optimal place.

2.1 Analysis of the Bees' Behavior

We performed experiments to investigate the aggregation behavior of young honeybees. A cohort of young (1 day old) bees was introduced into an arena consisting of a comb plate (floor) and four walls. The apparatus was set up in a dark room. A temperature gradient was established using an IR-lamp (bulb) that was shielded by a SCHOTTfilter. Such a filter allows only red and IR light to pass through. These wavelengths are invisible to honeybees, so they were navigating only in the temperature field and had no visual cues. An IR-sensitive camera was mounted on top of the arena. We report only selected data on the behavioral analysis of the bees, because our goal is to demonstrate the re-embodiment of the bees' individual behavior. The full analysis of the global patterns and of individual behavior will be published separately.

A representative pattern emerging in the temperature arena is depicted in Figure 1. For the following analysis, we defined four concentric zones that were arranged around the point of optimal temperature. The arrangement of these zones is depicted in Figures 2 and 3. Zone 1 was located directly below (and around)

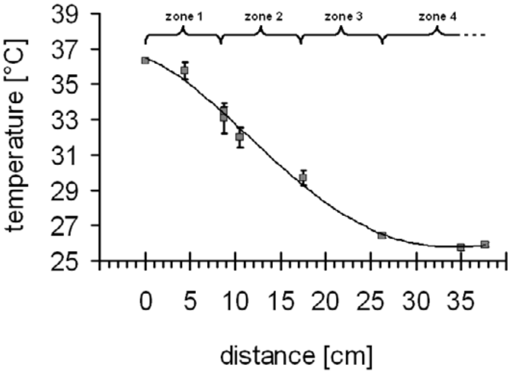


Figure 2 Measurements of the temperature field in the bee arena.

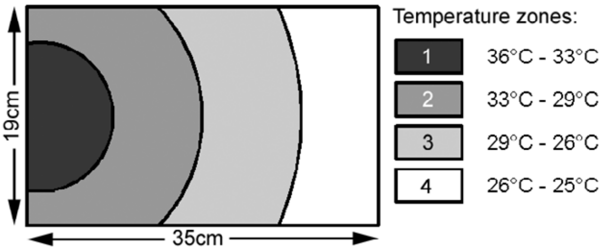


Figure 3 The temperature measurements divide the area into four distinct concentric temperature zones.

the lamp, it had the highest average temperature (34.5 ± 1.5 °C). Zone 4 was located at the remote end of the arena and had the lowest average temperature, which was 25.5 ± 0.5 °C.

The analysis of the emerging aggregation pattern showed that more clusters were formed in those zones that were near the optimum point. Near the optimum place in the arena larger clusters were formed and clusters survived longer than in remote areas of the arena (see Figure 4).

We also analyzed individual behavior of bees. We observed nine randomly selected bees for 10 min and recorded all behaviors (walking, resting) and all events (collisions) that happened during the observation using NoldusObserver.

For the re-embodiment of the honeybee behavior, the first question we had to answer was: Are the bees able to discriminate between a collision with a wall and a collision with another bee? To analyze this we browsed the observation-logs for all collision events and classified these events into four categories: (a) collision with the wall and stop of motion afterward, (b) collision with the wall and moving on, (c) collision with another bee and stop of motion, and (d) collision with another bee and moving on.

As illustrated in Figure 5, the bees were able to discriminate between other bees and walls, because they almost always stopped next to other bees; when a bee collided with a wall, it did not stop there.

The final question was: Why are clusters at lower temperatures (far away from the optimum) shorter lived than clusters near the optimum? Our assumption was that bees have two variables that they optimize: social contact and temperature. We also assumed that the resting time of a bee in a cluster depends on local temperature. Analysis showed that the mean resting time of bees was indeed longer in the warmer zones than in the colder zones (see Figure 6).

Based on these findings, we can summarize the core behavioral algorithm of an individual bee as depicted in Figure 7.

We assumed that the reason for the non-directed random walk of bees in our temperature field was that a flat gradient could not be exploited by a single bee when moving uphill in the gradient. Perhaps the temperature difference was too small at the bees' sensors,

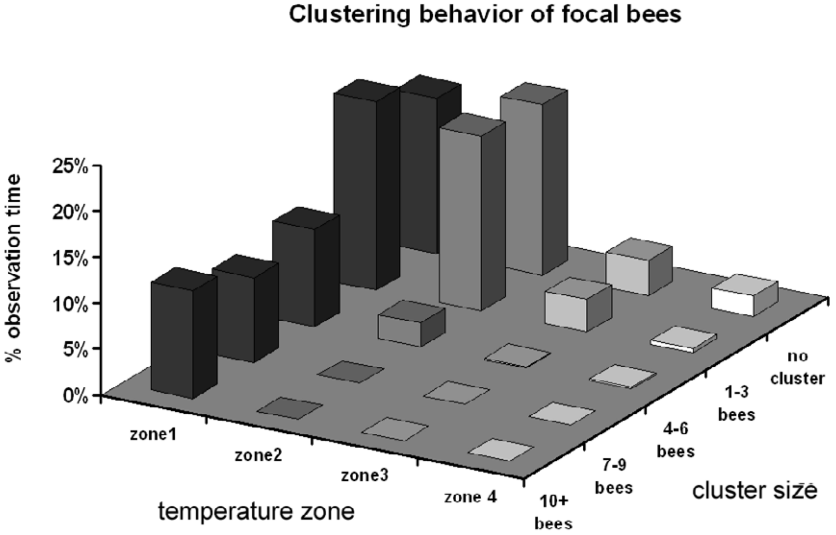


Figure 4 Clustering behavior of focal bees (N = 9 bees). We recorded the fraction of observation time that was spent in a cluster by each focal bee. These data were classified into the four temperature zones and by social contact, that is, the size of the cluster the bees were located in.

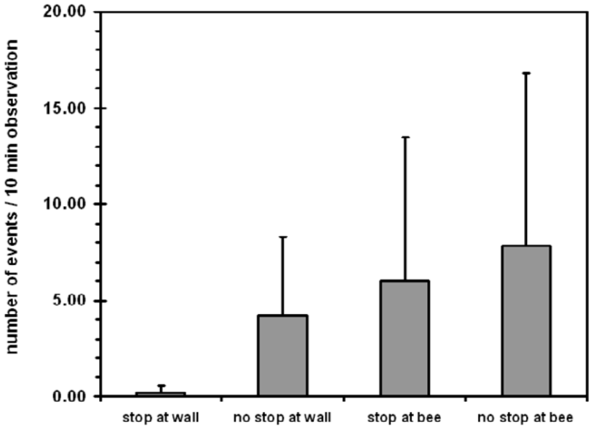


Figure 5 Resting of bees at walls and resting of bees with other bees. *N* = 9 observed bees. Bars indicate mean values, whiskers indicate standard deviations.

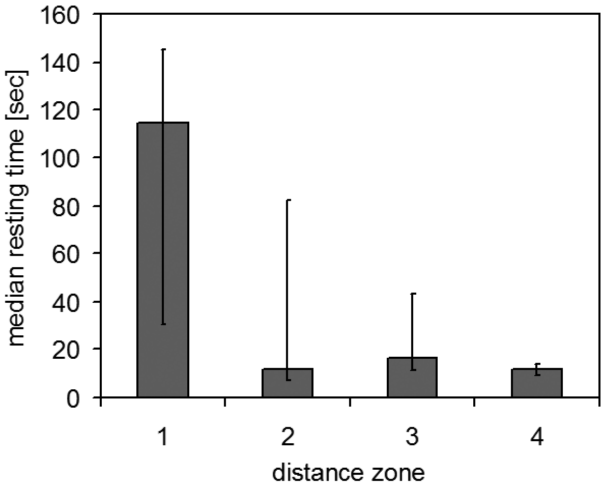


Figure 6 Median and inter-quartile ranges of observed resting times of bees in the four defined temperature zones. *N* = 9 observed be

which are located mainly in their antennae. The bees move almost randomly and often stop when they hit another bee. In this way a cluster is formed and by chance (depending on the density of bees) other bees can join the cluster. The bigger the cluster is, the more likely free-running bees are captured by the cluster and the more unlikely it is that the cluster will decrease to a size of one. In addition the probability that bees leave a cluster depends on the local temperature and not on the local gradient. The warmer it is, the longer a bee stays in the cluster. Thus, in the initial phase, many clusters are formed, but over time, sub-optimal clusters shrink and near-optimal clusters grow. Finally, only one cluster remains at the optimal position in the arena.

Next we implemented a distributed algorithm in a swarm of mobile robots by using a light gradient instead of a temperature gradient as the environmental template. We assumed that the resulting collective behavior of the robot swarm would show the following properties, according to the principles mentioned in Kennedy and Eberhart (2001) and Millonas (1994).

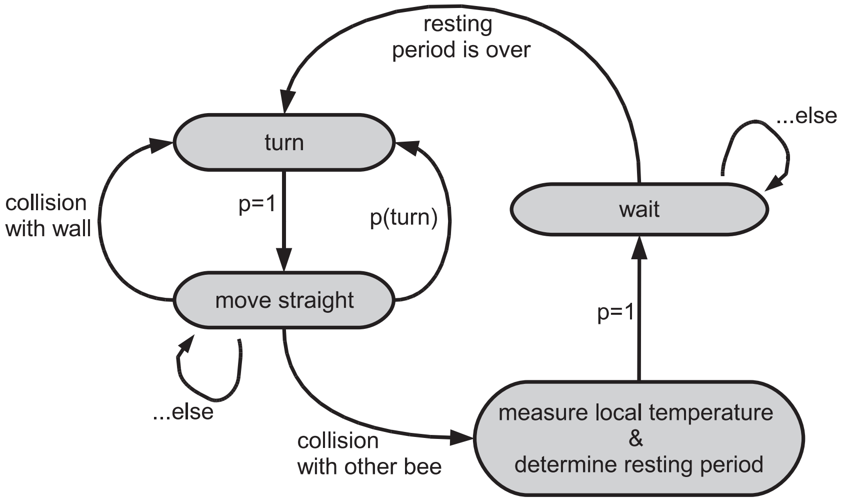


Figure 7 Finite state automaton that can describe the observed bee behavior.

Stability. The swarm finds a stable final solution. The solution does not depend on the initial distribution of robots in the arena. In addition, the measurement of the sensory data is only performed when the robots are standing in clusters. In cases where measurement is very noisy,[1] the measurement is performed many times and then averaged. In the hypothesized algorithm, no sensor data (except collision data) is used during the moving phase of the robot.

Flexibility. The collective decision of the swarm is flexible, so after a translation of the place of optimal light, the robots will collectively find the new optimum.

Scaling properties. The intelligence does not reside within one single individual. One individual alone cannot find a solution using this algorithm. The more individuals that perform this behavior simultaneously, the faster and the more precisely the optimum solution is found. The suggested algorithm works better with higher numbers of bees than with lower numbers of bees. An algorithm that depends on comparisons performed by individuals (such as a greedy uphill walker), will work most efficiently if performed by one agent alone. A collective-based algorithm should work better if performed by many agents in parallel.

Computational effort. Other algorithms will require much more computational effort and sensory abilities within each single agent. For example, three possible simple alternative algorithms for the same problem are discussed (Mletzko, 2006), that all work purely within each robot and do not involve social interaction between the robots:

1. Greedy uphill walker: needs two sensors and a periodic assessment of two temperature sensors. In addition, a control mechanism is needed that steers the robot in the appropriate direction.
2. A robot that has just one sensor and exploits the gradient: this robot requires a memory to store at least the last measurement and compare it with the current one. Afterward, a computation of the gradient with respect to the traveled distance can be performed to evaluate the gradient. Again, a steering algorithm to drive the robot uphill is also needed.
3. A robot that moves randomly and stops if the desired temperature is achieved. This robot would still need to evaluate the environment periodically and will need memory.

In contrast to our suggested bio-inspired algorithm, all three algorithms described above would additionally need efficient collision avoidance if they are being performed with several robots in parallel. The algorithm we propose achieves the same goal but without needing permanent updating of light-sensor data, an uphill-steering algorithm, memory, and a second sensor. The hypothesized algorithm minimizes sensor and computation because the local light sensor data are evaluated only once per collision with another robot. This is the reason the performance of the swarm increases with increasing density of robots in the arena: more frequent collisions lead to a higher update frequency of sensor data during the initial phase and the environment is collectively scanned by the swarm in smaller area intervals. After the initial clusters are formed, the collision frequency decreases automatically and so the computational effort is also minimized. After environmental fluctuation, the robots move around again, as the previous cluster dissolves and a new one forms, thus requiring less computational effort.

3 Re-embodiment of Individual Behavior for a Micro-Robot

3.1 Description of the Jasmine Micro-Robot

For performing the swarm experiments and testing the embodiment concept we used the Jasmine micro-robots, see Figure 8. It is a public open-hardware development with the goal of creating a simple and cost-effective micro-robotic platform and facilitating knowledge exchange in the swarm-robotics community available at www.swarmrobot.org.

The micro-robot is 30 × 30 × 20 mm and uses two Atmel AVR Mega micro-controllers: Atmel Mega88 (motor control, odometry, touch, color, and internal energy sensing); and Mega168 (communication, sensing, perception, remote control, and user-defined tasks). Both micro-controllers communicate through a high-speed two-wired TWI (I2C) interface. It has on board 24 KB flash memory for program code, 2 KB RAM for data, and 1 KB nonvolatile memory for saving working data.

The robot has six (60° opening angle) communication channels (also used for proximity sensing) and

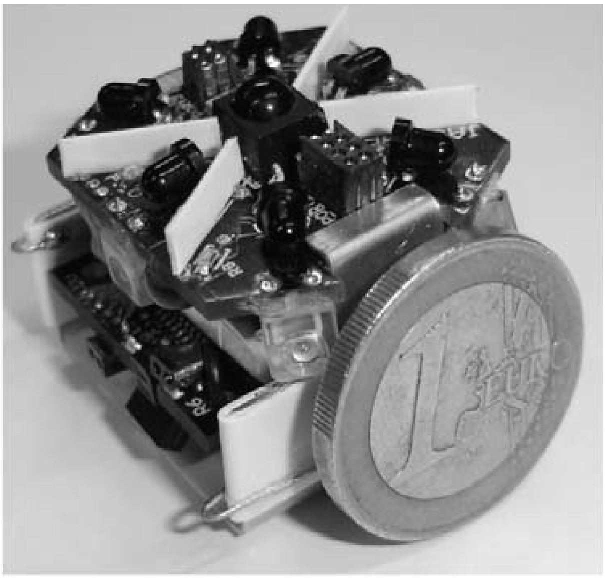


Figure 8 The third version of the Jasmine micro-robot

one geometry-perception channel (15° opening angle) based on separate IR receivers and transmitters. The communication area covers a 360° rose-like area with maximal and minimal ranges of 200 mm and 100 mm respectively. The physical communication range can be decreased through a change of sub-modulation frequency. The robot also has a remote control and robot–host communication (up-link and down-link), which is isolated from all other channels (through modulation).

The robot uses two DC motors with internal gears and two differentially driven wheels on one axis with a geared motor–wheel coupling. An encoderless odometrical system normalizes the motion of the robot (the robot is able to move straight forward and backward), and estimates the distance traveled with an accuracy of about 6% and a rotation angle of 11%. Jasmine III uses a 3 V power supply (from 3.7 V Li-Po accumulator) with internal IC-stabilization of voltage. Power consumption during motion is about 200 mA, when stationary about 6 mA, and in stand-by mode less than 1 mA. Power lasts from 1 to 2 hr during autonomous work. The robot is also capable of autonomous docking and recharging, so that the real time of experiments is effectively unlimited.

Robots are programmed using C with an open-source *gcc* compiler. There is a complete BIOS system that supports all low-level functions. Moreover, for quick implementation of swarm behavior there is a Jasmine-SDK system that includes an operational system and high-level functions based on Petri nets, see Figure 9.

Basically, the operational system executes four steps repeatedly: read sensor data, perform communication, make decisions and, finally, execute a plan. The interruption service takes care of software and hardware interruptions. The plan that a robot has to execute represents a Petri net consisting of two parts: a service part (handlers for interruptions) and a user-defined part (behavioral program for the robot). The structure of the service part represents an interruption-vector system with corresponding handlers. The interruptions (such as touch or low energy) are generated by the BIOS system, users only need to write the corresponding handlers. For further details of construction and programming (see swarmrobot.org or, for example, Kornienko, Kornienko, and Levi (2005b)).

3.2 Re-embodiment of Biological Strategy

For re-embodiment we used a four-step methodological approach, sketched in Figure 10.

Firstly, we try to formalize the sensor–actor capabilities of natural agents (bees), especially those which are assumed to be relevant for aggregation. The observed aggregation behavior of these agents is then expressed in terms of their sensor–actor capabilities, in other words, expressed as meta-models. Even in this early step we have to take into account realistic capabilities of artificial agents.

In the next step, the sensor–actor capabilities of natural agents are mapped into available sensor–actor capabilities of artificial agents. Here we have to take into account technological restrictions, such as characteristics and functionality of sensors, degrees of freedom (DOF) and holonomicity of the motion system, available energy, and so on. This step represents a virtualization of sensing and actuation. For example, when some required sensor capabilities cannot directly be reproduced, they can be indirectly obtained by other sensors and adapted to the required ones by software.

The next step represents a trade-off between the virtualized sensor–actor capabilities of artificial agents and the original behavioral strategy of natural agents. Obviously, not all required sensors and actors can

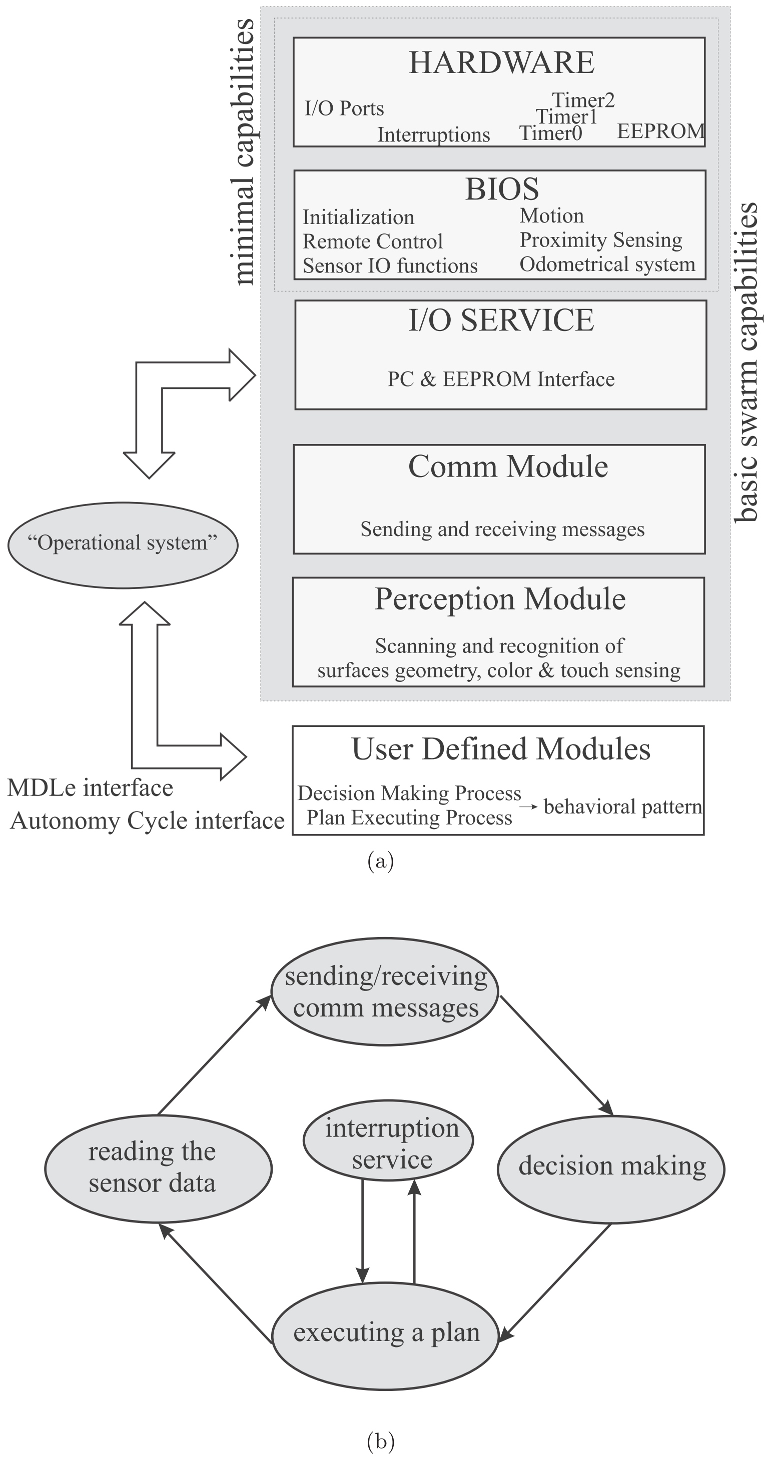


Figure 9 (a) Structure of Jasmine-SDK; (b) Operational system of the robot.

even be virtually implemented, so that an adaptation of the original strategy is required. However, this adaptation should be performed carefully so as to keep the structure of the bio-inspired algorithm, because this structure is expected to fit the requirements of collective systems best.

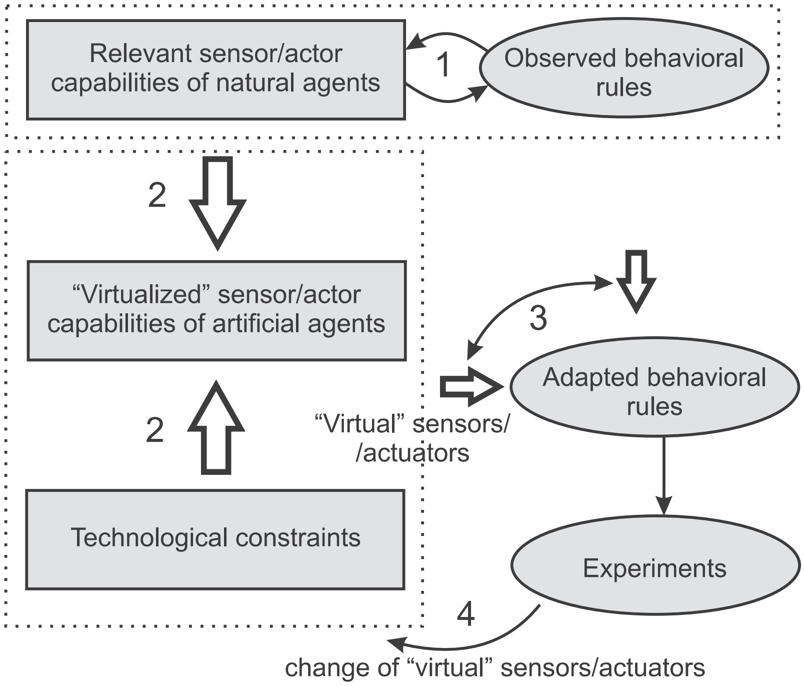

Relevant sensor/actor capabilities of natural agents
1
Observed behavioral rules
2
"Virtualized" sensor/actor capabilities of artificial agents
3
Adapted behavioral rules
2
"Virtual" sensors/ /actuators
Technological constraints
Experiments
4
change of "virtual" sensors/actuators

Figure 10 The structure of the re-embodiment procedure.

After the original behavioral strategy is implemented, experiments can be performed. In some cases (it happened in our re-embodiment experiment) the virtualized sensors–actors are still not suitable. In this case, a new implementation and virtualization of sensors and actuators should be performed until the emerged behavior becomes similar to the behavior exhibited by natural agents.

3.2.1 Representation of Behavioral Rules of Natural Agents in Terms of Their Sensor–Actor Capabilities For re-implementation of biological strategies, we first tried to understand the main differences between the observable behavior of bees and the known capabilities of micro-robots, as shown in Table 1.

As shown in this Table, there were several principal sensor–actor issues that made biological aggregation possible:

- Detection of temperature changes.
- Directional sensing of these changes (to detect a gradient).
- Detection of other bees and their differentiation from other objects.
- Behavior in dense clusters with a high degree of connectivity.

The idea of biological aggregation in terms of the bees' sensor–actor system is as follows: when the bees meet in a region with a temperature higher than ambient, they walk more slowly. The more bees that join a cluster, the more bees in this cluster are "blocked" by other bees and therefore the longer such a cluster can persist. The cluster is then available longer for other bees to join, which represents a positive feedback loop. In this way, the aggregation is a specific relationship between sensitivity of sensors and holonomicity of motion.

A topic assumed to have an essential impact on the global aggregation patterns is the building of *seed points*. Such seed points are initial aggregations of two bees that allow further growth of clusters. The two-bee clusters do not exist for long because bees are not blocked within them. Therefore, we assume the existence of a seed point factor: either the bee density for cluster formation should be much higher than in a normal case, or bees have a mechanism allowing two bees to stay together for a long time.

Table 1 Comparison between observable behavior of bees and known capabilities of micro-robots.

Bee behavior, sensor–actor system	Robot capabilities
Bees are able to differentiate between other bees and obstacles. This is achieved by cuticula-bound chemical substances (taste, smell).	This differentiation is achieved via emission of light signals by robots, which mimic the chemical substances on the bees' surfaces.
The duration of resting time in a cluster is a function of the local temperature.	The duration of resting time in a cluster is a function of the local illumination
Perception radius: bees detect other bees at close range or only by touch.	A robot can detect another robot by detecting its emitted IR-radiation. A robot cannot detect a contact with its chassis but this can be simulated by proximity sensing and rotation.
In flat gradients, the detection of the direction toward the temperature optimum seems to be impossible for a bee; it performs almost a random walk.	In robots, we do not measure the illumination during the movement phase. Our robots move straight and turn randomly after a collision with the border of the arena.
Bees are able to build a dense cluster with a high degree of connectivity.	Robots cannot build a dense cluster, because of the 2-DOF motion and the need of space for rotation.
Bees walk relatively slowly, 1–2 body lengths/s and the motion system can be described as holonomic.	A robot can change its velocity over a wide range; however, experiments should be done in low-velocity mode. The motion is non-holonomic.

3.2.2 Virtualization of a Sensor–Actor System in Artificial Agents Replication of the biological aggregation is not directly possible for the following reasons:

- The robots have no ability to sense temperature.
- These micro-robots cannot achieve clusters with high densities because of their sensor–actor capabilities; this fact can essentially change the behavior during a blocking phase and finally lead to cluster disaggregation.

These robot shortcomings can be compensated for by the following virtualization techniques.

1. Instead of temperature, we used a light gradient that approximately resembles the temperature gradient that was used in the bee experiments. However, the light intensity differs in several ways from temperature: light in not cumulative and the light intensity has a different distribution on the surface than temperature. This point is demonstrated in Figure 11a in the preliminary experiments.

   To allow a few robots to aggregate, the light region should be large enough and should not contain IR spectra. This is why we used a special round luminescent lamp for generation of the light gradient. The light intensity in the brightest region (region A) is almost constant across the whole of that region. The highest variance in intensity is found in the transient region (region B), while the rest of surface (region C) again has only small changes in intensity. In this way, a light gradient can be established only in the small region B.

   For re-embodiment we need sensors that can detect light intensity with sufficient sensitivity, but without being saturated (to work equally linearly in bright and dark regions). For that we first used small 3 × 3 mm clear-epoxy encapsulated solar cells, generating about 0.5 V, 1.9 mA in sunlight, see Figure 12a. They were connected directly to the ADC port of the microcontroller. To implement a gradient-search strategy using one light

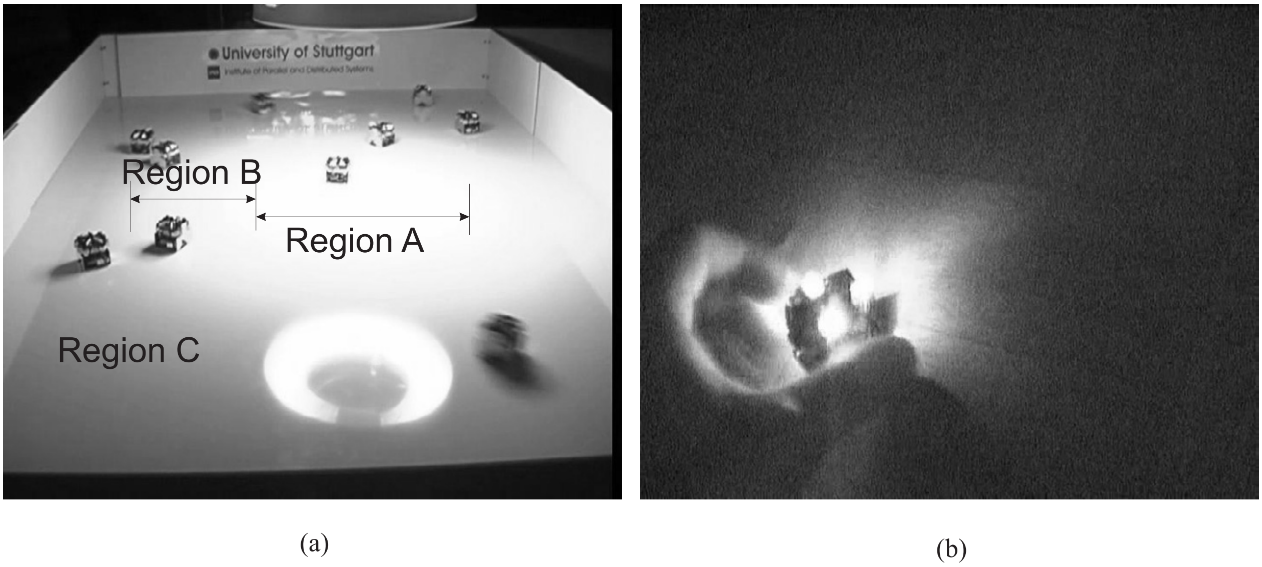


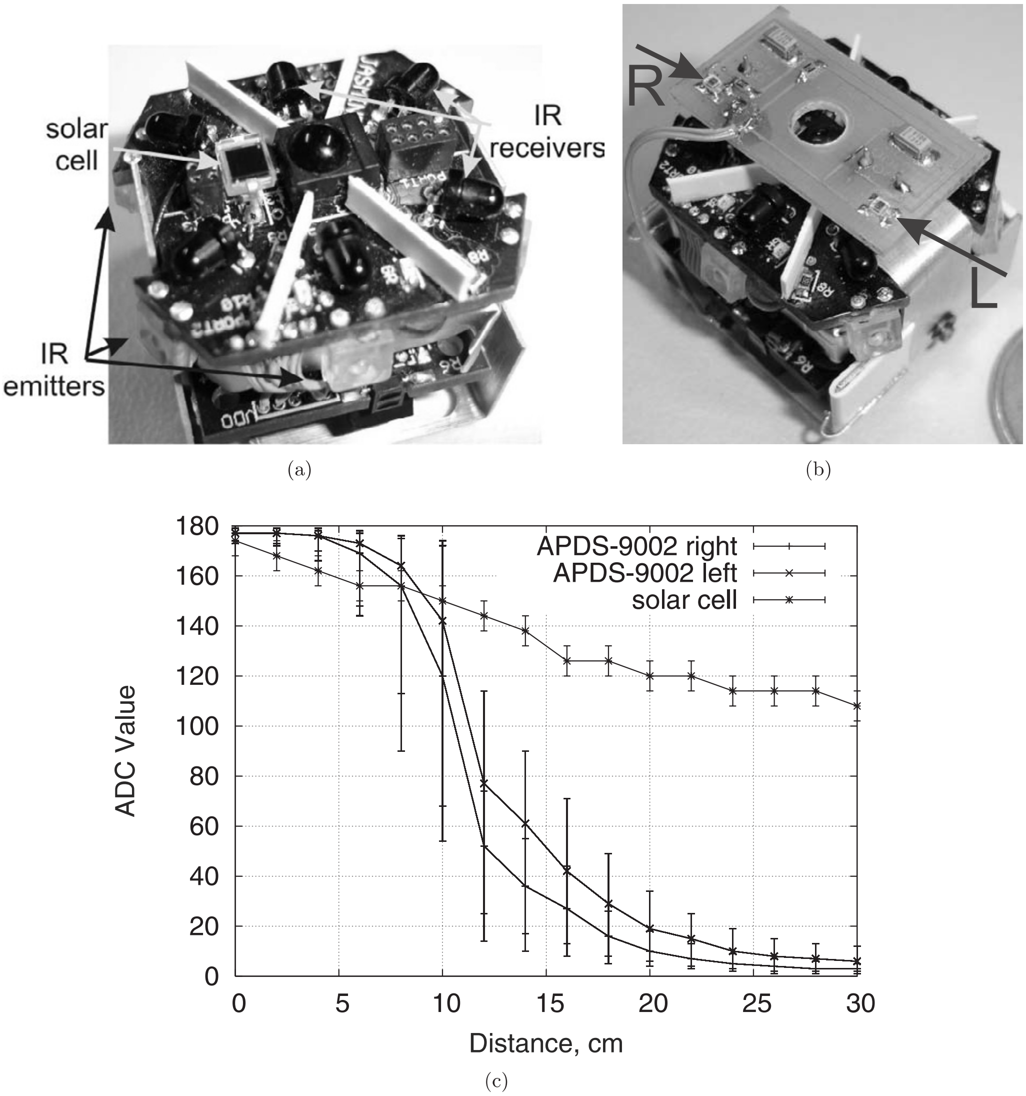
Figure 11 (a) Light gradient in the preliminary robotic experiments; (b) IR-light emitted by robot during proximity sensing (image is taken by IR camera).

sensor, the robot can rotate in the radius of approximately 5–7 cm and save the values of light intensity. However, after performing the preliminary experiments we arrived at the conclusion that this sensor, despite the linearity of the light–signal transformation, cannot be used in aggregation experiments. The reason was the low sensitivity to light changes, especially in dark regions. For example, the robot was not able to identify the region B, shown in Figure 11a. Therefore, after the preliminary experiments, we decided to change the light sensors to the more sensitive type APDS 9002, produced by Agilent Technologies; we installed two of them, on a small extension board (Figure 12b), for further experiments with gradient light. The comparison of light–signal conversion for both sensors is shown in Figure 12c. The solar cell is much more linear and inertial for the light changes, whereas APDS 9002 is more sensitive and has a faster reaction time. This sensor is able to detect light changes from an AC luminescent lamp, so we installed a passive RC-filter on the extension board and averaged the obtained values (dispersion of APDS 9002 is much higher than the solar cell as shown in Figure 12c).

2. To implement the virtual sensor, capable of distinguishing between objects and other robots, we used the following strategy. The robot emits the IR light for proximity sensing and perceives the IR-light reflected from obstacles in six channels (see Figure 12a). This emitted IR light can be sensed by a robot only at a short distance and can be interpreted as a “contact pheromone,” see Figure 11b.
   When a robot detects IR-signals, it assumes that some robots are in its local neighborhood. The robot then emits signals identifiable by other robots. This strategy underlies the virtualized robot-recognition sensor, for which the algorithm is shown in Figure 13.
   In this way, by a combination of existing sensors, specific behavior, and signal processing we are able to make a virtualized sensor of the required functionality.

3.2.3 Adaptation of Original Behavioral Rules to Virtualized Sensor–Actor Capabilities of Artificial Agents The behavioral pattern, shown in Figure 7, can be adapted to other motion capabilities of the robot (in the cluster-building part, especially seed points) and to other sensing capabilities (primarily the waiting parameters) as follows.

1. For adaptation of the algorithm for establishing a seed point in robotic experiments, the robots have



(a)

(b)

(c)

Figure 12 (a) The first version of the light sensor (solar cell) installed directly on the robot. Also shown are IR emitters and receivers. (b) The second version of the light sensor (sensor APDS 9002), installed on the robot as an extension board. (c) Sensitivity and nonlinearity of the first and second version sensors in the light–signal transformation. The light at different distances from the luminescent lamp was measured (as shown in Figure 11a, the distance between the arena surface and the lamp was 12 cm and ambient light was off); the values of the solar-cell sensor are multiplied by 6. In the signals from APDS 9002 we can easily recognize the corresponding regions from Figure 11a.

to meet and stay together long enough to allow other robots to join the cluster. We can analytically calculate this relation by using several analogies to the molecular-kinetic theory of ideal gas, such as diffusion in ideal gas. We introduce the following notation: the sensing radius R (R_s, R_c are collision avoiding and communication radii); l_c the length of free path from the start of motion to the first communication contact; l_s the length of free path from the start of motion to the first collision-avoiding contact; n_c and n_s are the number of communication and collision-avoiding contacts respectively; S_c and S_s are the areas of the "broken" rectangles created by the motion in time interval t with R_c and R_s; and finally the robot's motion velocity v. We do not consider collision avoidance caused by obstacles, changes in the robot's velocity or behavior during collision

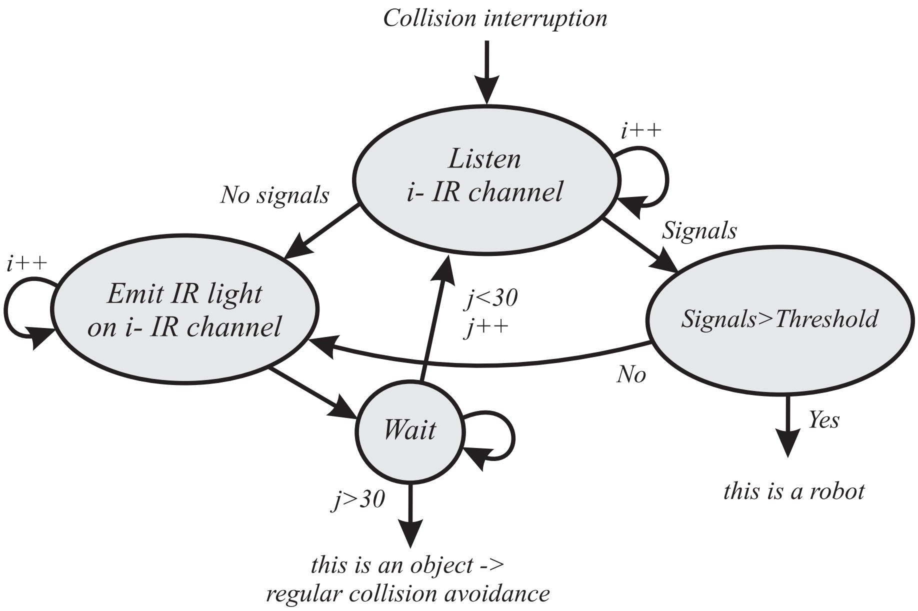


Figure 13 Algorithm of the virtualized sensor that is able to differentiate between another robot and a passive object without communication between robots.

avoidance. This simplifies calculations and we expect that this approximation error is not large for small clusters. Figure 14 illustrates this idea.

We differentiate between communication contacts and collision-avoiding contacts. For further calculation we use collision-avoiding contacts because they correspond more closely to the bio-inspired approach. Firstly, we are interested in the number of two-robot contacts, n_s, that happen during the movement. This value is equal to the average number of robots in the area S_s,

$$n_s = S_s D_{sw}, \tag{1}$$

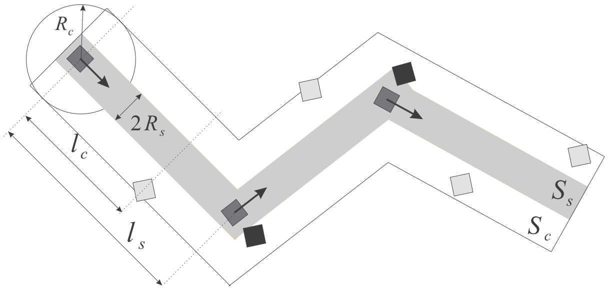


Figure 14 Motion path of a robot with communication and collision-avoiding contacts.

where D_{sw} is the swarm density. We assume that the robot's rotation radius is small (robots can rotate in one place), so that we can neglect the area of fractures. In this case $S_s = 2R_s\upsilon t$. D_{sw} can be calculated as the number of robots, N, in a swarm divided by the area available for the whole swarm ($S_{sw} - NS_r$):

$$D_{sw} = \frac{N}{S_{sw}} \rightarrow n_s = \frac{2R_s\upsilon tN}{S_{sw} - NS_r}, \tag{2}$$

where S_{sw} is the whole area and S_r is the area occupied by robot itself. In relation 2 we assume only one robot moves while the others are motionless. The more exact relation for the case when all robots move differs from 2 only by the numeric coefficient $\sqrt{2}$ (as proved by Maxwell for a diffusion in ideal gas). For further calculation we use

$$n_s = \frac{2\sqrt{2}R_s\upsilon tN}{S_{sw} - NS_r} \rightarrow t = \frac{(S_{sw} - NS_r)n_s}{2\sqrt{2}R_s\upsilon N}. \tag{3}$$

Obviously, the two-robot contact ($n_s = 2$) will happen during the time t_2 and after this, the

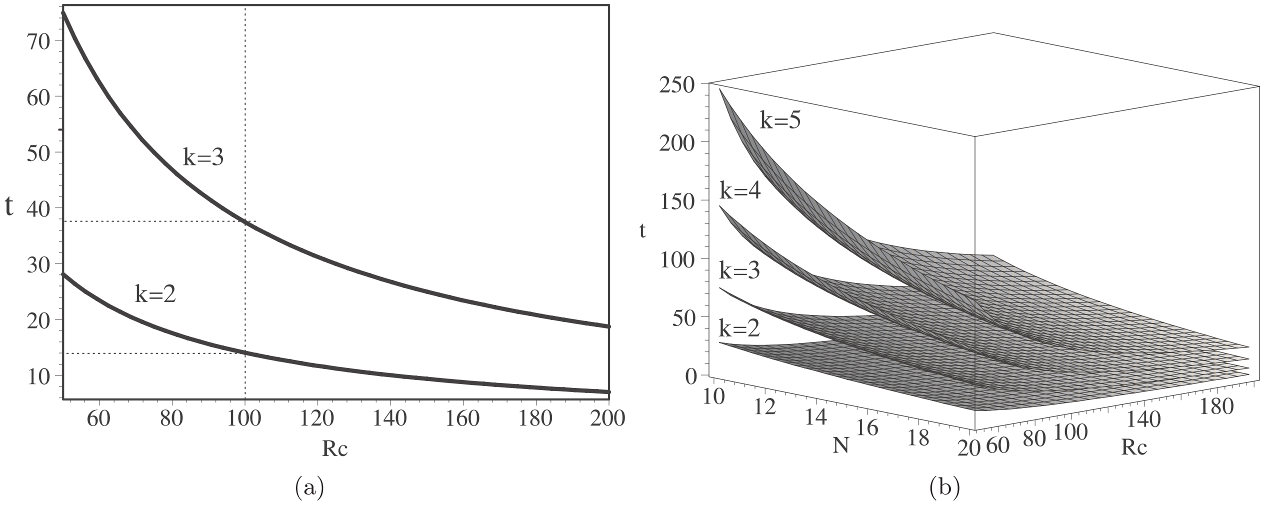


Figure 15 (a) Plot of Equation 4; dependencies between t and R_c for $k = 2,3$. (b) Plot of Equation 4; dependencies between t, R_c, and N for $k = 2 \dots 5$. R_c is calculated in mm, t in seconds.

number of available robots is decreased by one (after contact the two robots stop and are considered as one robot of double size). The time for the next contact can be calculated by using relation 3 when $N = N - 1$; we are interested only in a small number of total contacts (when the number of robots in a cluster is smaller than the number of available robots, we can skip changes in behavior caused by non-holonomic motion). We can express the total time t_k of k-contacts ($N/2 > k > 2$) as

$$t_k = \frac{S_{sw} - NS_r}{2\sqrt{2}R_s\upsilon} \sum_{i=2\dots k} \frac{i}{N-i+2}. \tag{4}$$

For the calculation, we took $S_{sw} = 1{,}000 \times 1{,}000$ mm^2, $S_r = 26 \times 26$ mm^2, $R_s = 100$ mm, $\upsilon = 50$ mm s^{-1}, and $N = 10$. For two-robot contact we get typical times for $t_2 \simeq 14$ s, $t_3 \simeq 37$ s, $t_4 \simeq 72$ s, that is, robots do not move 23–35 s and wait until the next robot comes into the cluster. The dependencies between t, R_c, and N for different values are shown in Figure 15. These periods agree with experimental data from the preliminary experiments with micro-robots.

As indicated by the calculation, the seed point for building clusters can appear when robots wait at two-robot contact for at least 23–35 s (this time can decrease with increasing swarm density). The waiting (resting) time matches very well with the sensor-related adaptation of the algorithm.

2. For adaptation of the original bee algorithm shown in Figure 7 for other sensors, we measured the light gradient in the robot arena (as we measured the temperature gradient in the honey bees). Figure 16 shows the results of these measurements. The light gradient in the arena (Figure 16) is very similar to the temperature gradient in the honeybee arena (Figure 2). The main difference is that

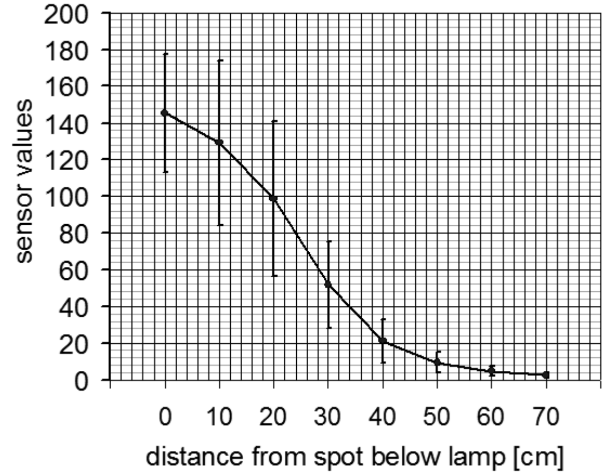


Figure 16 Light gradient in the robotic experiments arena. Dots indicate mean values (N = 12 per measurement), whiskers indicate standard deviations. The amount of noise in the sensor data is clearly visible.

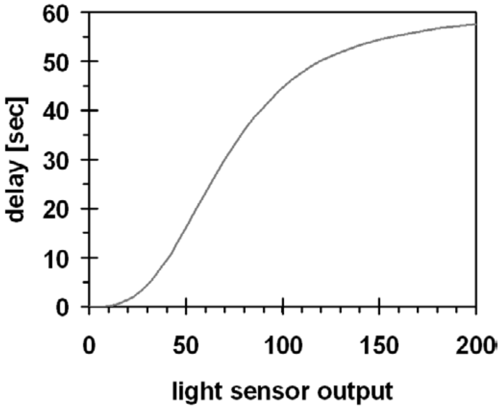


Figure 17 The stimulus–response curve we used in our robot experiments. The higher the sensory values of the light sensor were, the longer the robot waited in a cluster.

the temperature scaled between 25 °C and 36 °C while the measurable light intensity values were between 0 and 145. For a successful embodiment, we developed a conversion rate between these two measurements (see Equation 5):

$$temperature_{local} = 0.0759 * light\_value + 25.0. \quad (5)$$

The duration of the bees' stays in the temperature zones (see Figure 6) showed a clear nonlinear effect with the longest resting times in Zone 1. To incorporate this into the robot algorithm, we implemented a nonlinear stimulus–response curve, which is depicted in Figure 17.

4 Implementation of Aggregation Strategy for a Micro-Robotic Swarm and Performing the Experiments

The structure of the algorithm for robot behavior is shown in Figure 18.

After the robot encounters an obstacle, the corresponding interruption is generated. A robot attempts to determine whether this obstacle is another robot or some passive object. To do this, it waits approximately 200 ms for the collision-avoidance signals of other robots in the environment. These received IR-signals indicate the local presence of another robot (the robot itself also emits such signals during the detection phase). When there are no such signals a passive object, which cannot emit IR signals, is indicated. When the collision contact is another robot, the local light intensity will be measured. The robot reads several values from both sensors and then averages them. Depending on the value received (through interruption), the robot sets its timer according to the curve depicted in Figure 17 to wait for 50–60 s without motion, but still emitting IR signals to be identifiable as a robot. This allows the creation of a seed point for further clusters. After this time the robot turns back to the cluster/robot and starts a moving again. When there are other collision contacts, the whole procedure is repeated. In this way, the robot remains blocked within the cluster.

We performed two series of experiments: the preliminary experiments, where we looked for appropriate sensors (see Figure 11a) and the final experiments, shown in Figure 19.

The final experiments included a series of trials with 1–18 robots, equipped with the extension sensors board, as shown in Figure 12b. As in the preliminary experiments, we used a round luminescent lamp, which was mounted about 40–50 cm above the arena. The ambient light was also generated by tube-luminescent lamps about 3 m above the arena. The size of arena was 140 × 115 cm. Figure 20 illustrates the images from an experiment using three robots. It is clear that there was no clustering at the target location. In follow-up experiments, we increased the number of robots step-by-step. The critical swarm density, where aggregation at the target place was initiated, was reached with nine robots. In Figure 21 we show pictures from an experiment with 15 robots in which the robots aggregated quickly at the target location. In further experiments we moved the lamp to different positions of the arena, the robots disaggregated and re-aggregated at the new location. A video of these final experiments can be downloaded from www.swarmrobot.org.

4.1 Aggregation Time and Scalability of Robot Behavior

During preliminary, final, and post-final experiments we performed over 100 trials (the final experiment with 18 robots was used as a demonstrator in a museum). Additionally, a few post-final experiments with an extra-large robot arena (3 × 3 m) and the number of robots 105, 75, 50, 35, and 25 were performed (Figure 22).

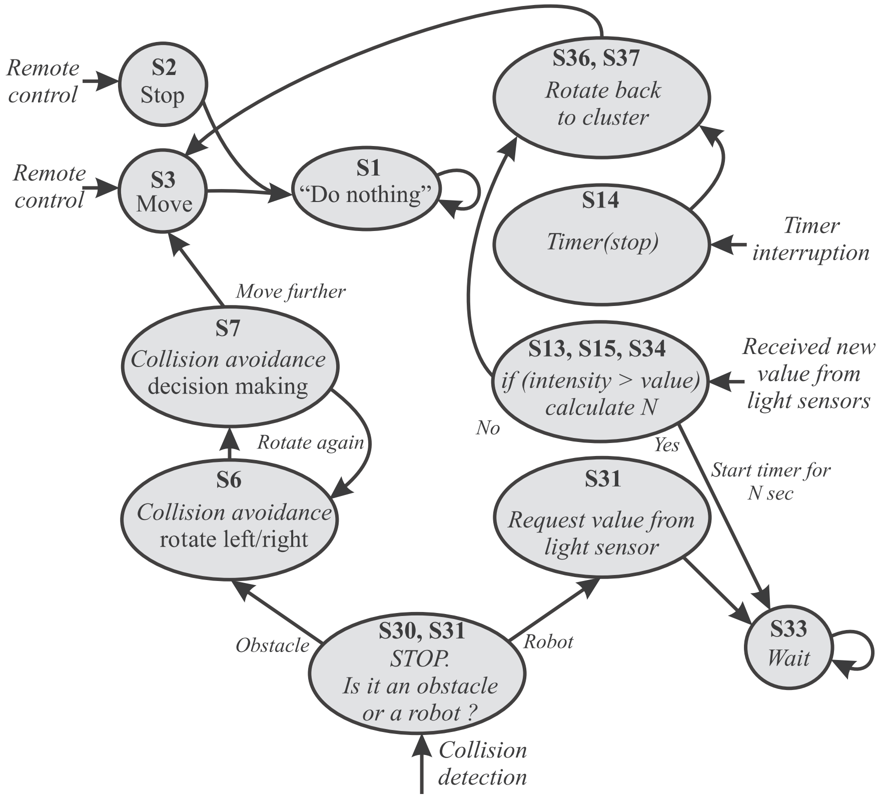


Figure 18 (a) The structure of the algorithm for robot behavior originating from the biological observation and adapted for robots sensor–actor system. (b) Large-scale swarm (max. 135 robots.)

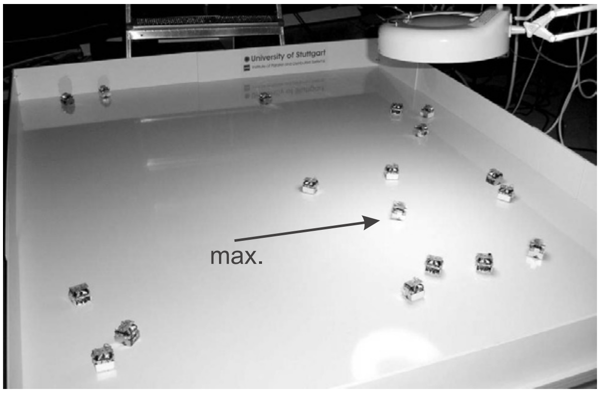
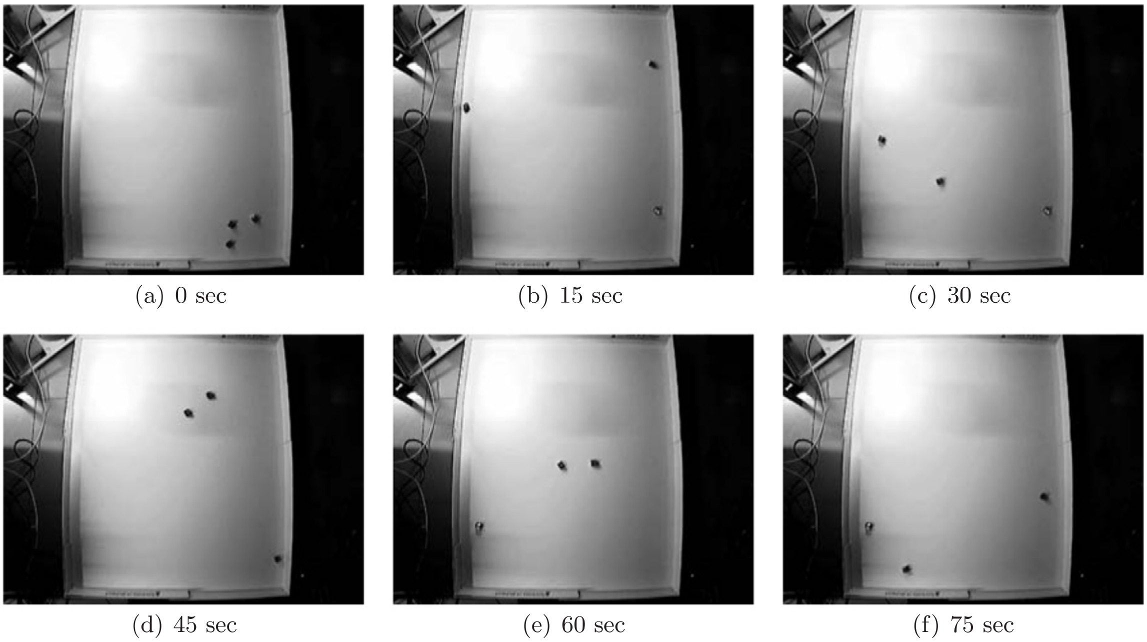


Figure 19 Final experiments with 1–18 micro-robots.

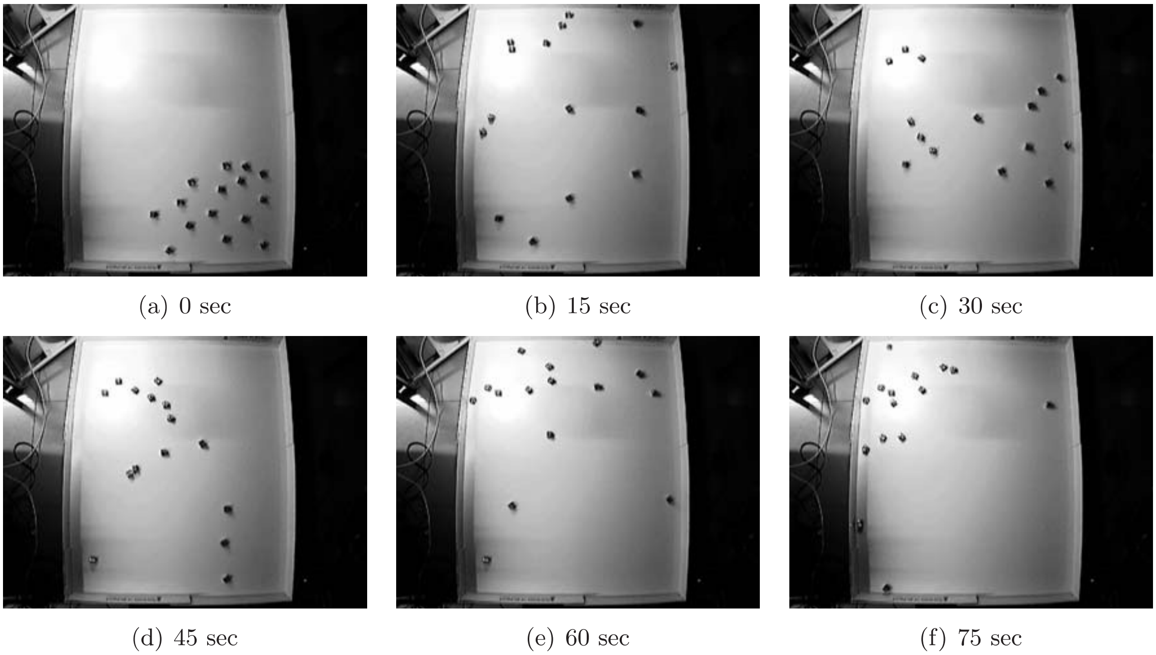
After performing these experiments we identified three main factors influencing the aggregation. The first is the number of collision contacts, n_s, defined by the relation between swarm density D_s, perception radius R_s, and velocity of motion υ, and expressed by relation 3. Increasing n_s (for example by increasing D_s, R_s, or υ) leads to faster aggregation. When n_s remains constant, performance of the aggregation is expected to also be constant (taking into account two other factors). The second factor is the area, S_{sp}, and the position of the light/temperature spot. We assumed that aggregation is successful when about 75% of all robots are located in one cluster under lamp(s). However, when $S_{sp} < 0.75NR_s^2\pi$, robots do not have enough space for aggregation and this factor could be an environmental bottle-neck. In this case we count aggregation as successful when the area under lamp is occupied by robots even when < 75% of all robots are located there. The position of the light spot on the arena defines how quickly the seed point can be created. For example, 18 robots require about 1 min for aggregation with the light spot in the corner area. In contrast, 15 robots require only 30–40 s with the light in the middle of the arena. The last factor, which influences the performance and scalability, is the waiting time in clusters. Robots situated in the middle of a



(a) 0 sec (b) 15 sec (c) 30 sec

(d) 45 sec (e) 60 sec (f) 75 sec

Figure 20 A group of three robots navigating with our algorithm in the arena. They were not able to find the place with the optimal illumination, which was located in the upper left corner of the arena.



(a) 0 sec (b) 15 sec (c) 30 sec

(d) 45 sec (e) 60 sec (f) 75 sec

Figure 21 A group of 15 robots navigating with our algorithm in the arena. They quickly and collectively found the optimal spot, which was located in the upper left corner of the arena.

cluster are blocked by other robots and therefore remain in this cluster. However, robots in boundary areas can easily leave the cluster and this leads to disaggregation. When the waiting time is too short, small

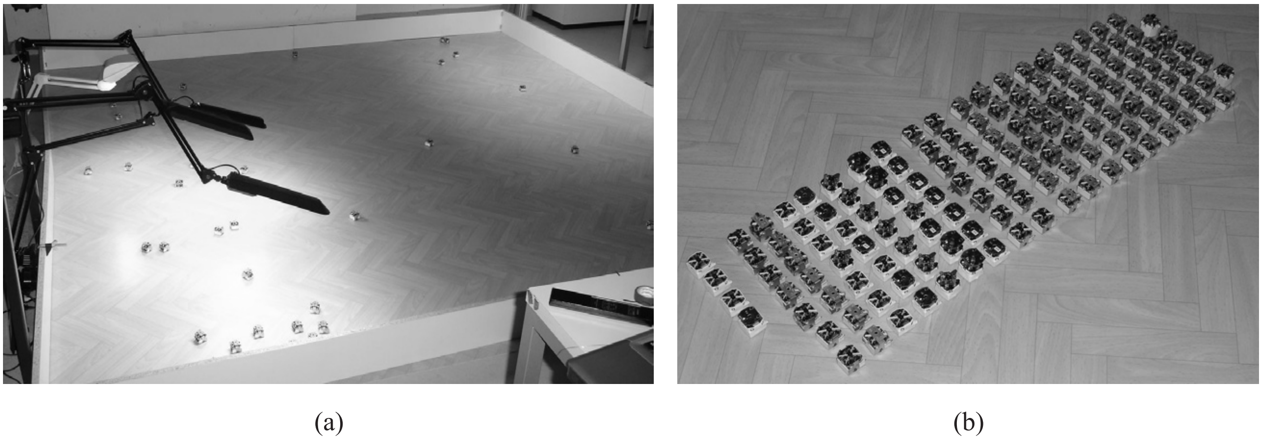


Figure 22 (a) Extra-large robot arena (3 × 3 m) used in post-final experiments; (b) Large-scale swarm (135 robots) in post-final experiments.

initial clusters will always disaggregate, so that robots do not find an optimum. When the waiting time is too long, an optimal cluster will also not appear because non-optimal clusters will not disaggregate. The maximal waiting time is experimentally chosen as 1 min.

The scalability of bees aggregation behavior was tested by performing the experiment with 16 and 32 bees for the same arena conditions and temperature regions. For the case of 16 bees the averaged aggregation time is about 128 s. For the case of 32 bees, the area of temperature Zone 1 (see Figure 2) seems to be too small for the quota of 75%. Bees are aggregated very closely to each other (clearly visible in Figure 1) so that the approach expressed by relation 3 cannot be applied to the estimation of scalability. For example, the quota of 75% for 32 bees corresponds to the aggregation time of 217 s. We can see that there is something wrong with this value, because increasing the number of swarm agents in a subcritical domain of swarm density should improve the performance, that is, it should be < 128 s (as demonstrated by the robotic experiments). Therefore, we consider success here as the quota of ≈ 33% (≈ 69 s), which approximately corresponds to 75% in the case of 16 bees. Both values are shown in Table 2.

To test the scalability of the algorithm, the number of robots (18, 25, 35, 50, 75, 105), the area of the light spot (S_{sp} = 4,025 cm^2, 11,250 cm^2, 22,500 cm^2) and robot arena (S_{sw} = 140 × 115 cm^2, 300 × 300 cm^2) are increased step-wise. We repeated each experiment on a small arena S_{sw} = 140 × 115 cm^2 7–8 times and on a large arena S_{sw} = 300 × 300 cm^2 at least 2–3 times, and calculated averaged aggregation time for each type. Although this number of repeats is statistically low, large-scale experiments are difficult to repeat for technical reasons.

For 15–18 robots in an arena S_{sw} = 140 × 115 cm^2 and S_{sw} = 4,025 cm^2 light spot the aggregation time was 30–70 s in almost all experiments (depending on the position of the light spot, averaged 65 s). The aggregation time for 25 and 35 robots in the same arena and light spot is decreased to 25–40 s (however in this case > 25% of robots moved outside of the light spot). The aggregation time for the arena size 300 × 300 cm^2 with 35 robots and area of the light spot S_{sp} = 11,250 cm^2 was about 2.5 min. Further increasing the number of robots (50, 75, 105) in the same area of robot arena and light spot sped up aggregation to 40 s (however, here also > 25% of robots moved outside of the light spot).

The area occupied by aggregated robots is usually larger than expressed by $0{,}75NR_s^2\pi$, because the robots build clusters of arbitrary geometrical forms. We can estimate this area as approximately double that expected. Therefore, the large number of robots (50, 70, 105) occupied about 15,079 cm^2, 21,111 cm^2, and 31,667 cm^2 respectively. This is much larger than the spot area. We therefore decided to double the area of the light spot (S_{sp} = 22,500 cm^2) by using additional lamps, see Figure 22a. In this way the area of a light spot was about 25% of the whole arena. This led to faster initiation of seed point(s) and faster aggregation for small N; however, the averaged aggregation time

Table 2 Collection of experimental data from all experiments: S_{sw}, arena size, m^2; S_{sp}, spot size, m^2; S_r, size of the robot/bee, mm^2; R_s, perception radius, mm (perception area of a bee is about 17 × 8 mm, we approximate this as a perception radius 6 mm in sense of (3)); υ, velocity of motion, mm s^{-1}(for bees 6–60 mm s^{-1}, for robots 250–300 mm s^{-1}; we used maximal values for further calculations).

Bees					N of Bees/Aggregation time (s)					
S_{sw}	S_{sp}	S_r	R_s	υ	16	32	32			
0.0665	0.01662	66	6	60	128_s	$69_{(\approx 33\%)}$	$217_{s\ (75\%)}$			
Robots					N of Robots/Aggregation time (s)					
S_{sw}	S_{sp}	S_r	R_s	υ	15	25	35	50	70	105
1.61	0.4025	900	60	300	65_s	38_s	25_s			
9	1.125	900	60	300			148_s	98_s	73_s	48_s
9	2.25	900	60	300			129_s	92_s	69_s	45_s

for large N remained similar to the value of a smaller light spot. The experimental data is presented in Table 2.

Using data from this table we can calculate several scalability metrics for different conditions. We already mentioned that one of the most important parameters is the number of contacts n_s, expressed by relation 3. We calculate n_s for the aggregation time in each case. When the behavior is scalable n_s should provide similar values for different experiments. The value n_s/t can be also of interest because it points to the number of contacts per second. Finally, when $S_{sw} \gg NS_r$ (the arena is much larger than area occupied by all robots), we use the swarm reactivity as a metric, expressed by

$$\varphi = \frac{2\sqrt{2}R_s\upsilon}{S_{sw}}. \qquad (6)$$

The value φ has a physical dimension $1/t$ and demonstrates internal inertness of the swarm. We summarize the values of n_s, n_s/t, and φ in Table 3.

We see that n_s fluctuates around the mean of 29.15. These fluctuations can be explained by a systematic inaccuracy in the experiments (e.g., estimation of aggregation time was made by an operator observing the behavior) or by variation of parameters in robots (e.g., robots do not always move with a constant velocity of 300 mm s^{-1}). However, this small fluctuation around 29.15 for very different S_{sw}, N, and t means a good scalability of the behavior.

The n_s/t demonstrates which systems are fastest and slowest; for example, the fastest system is the case of 35 robots in a small arena. It is interesting to note that the case of 16 bees is equivalent to the case 50 robots in a large arena.

Finally, the value of φ characterizes three different classes of swarm systems: bees in a small arena, robots in a small arena, and robots in a large arena. We see that inertness of robots and bees in a small arena is similar, it means that we can achieve similar performance of the system at similar input values. In contrast, the case of a large arena indicates 4–6 times worse performance of this system: in order to achieve the same value of n_s we need 4–6 times more robots (or more time).

5 Conclusion

We believe that the proposed algorithm is one of the simplest possible optimum-finding algorithms for swarm robotics (concerning algorithmic complexity, computational efforts and sensory abilities). In addition, this algorithm shows interesting scaling properties because the inter-robot collisions enhance collective performance in contrast to usual algorithms where such collisions are seen as counterproductive events. We investigated these scaling properties of

Table 3 Values of n_s, n_s/t, and φ in dependence of S_{sw}, N, and t for bees and robots.

	S_{sw}	N	t	n_s	n_s/t	φ	Remarks
1	0.0665	16	128	31.86	0.248	0.0153	bees
2	0.0665	32	69	34.91	0.506	0.0153	bees for ≈ 33% quota
3	1.61	15	65	31.09	0.478	0.0316	robots
4	1.61	25	38	30.46	0.801	0.0316	robots
5	1.61	35	25	28.22	1.128	0.0316	robots
6	9	35	148	29.40	0.198	0.00565	robots, $S_{sp} = 1.125$ m^2
7	9	35	129	25.63	0.198	0.00565	robots, $S_{sp} = 2.25$ m^2
8	9	50	98	27.85	0.284	0.00565	robots, $S_{sp} = 1.125$ m^2
9	9	50	92	26.15	0.284	0.00565	robots, $S_{sp} = 2.25$ m^2
11	9	70	73	29.11	0.398	0.00565	robots, $S_{sp} = 1.125$ m^2
10	9	70	69	27.51	0.398	0.00565	robots, $S_{sp} = 2.25$ m^2
12	9	105	48	28.81	0.600	0.00565	robots, $S_{sp} = 1.125$ m^2
13	9	105	45	27.01	0.600	0.00565	robots, $S_{sp} = 2.25$ m^2

aggregation behavior analytically and numerically. It was demonstrated that n_s (the number of contacts) is the most appropriate value to estimate the scalability class of swarm algorithms. For example, n_s varies around 29.15 for all performed experiments, indicating the super-scalable class (Constantinescu, Kornienko, Kornienko, & Heinkel, 2004) of this algorithm. The value of n_s can also serve as a performance metric (instead of pure aggregation time t) because it is independent of such factors as arena size, number of robots and others. We should also note several difficulties encountered in performing the experiments on scalability. For example, it is almost impossible to keep the same swarm density in the light/temperature areas for different N of robots/bees. This, with other factors, impacts accuracy of n_s.

The algorithm shows that the robot-to-robot interactions are essential for achieving the common goal, but the algorithm does not involve any robot-to-robot communication which means no messages are passed from one robot to another. So an almost paradoxical (at least a counterintuitive) situation emerges: a swarm of non-communicating robots works more efficiently in larger groups than in smaller ones. In addition, the algorithm has all the properties that characterize swarm intelligence. These include: emergence, flexibility, robustness against perturbation, robustness against initial conditions, and robustness against sub-optimal solutions (local optima).

Another important result is the systematic re-embodiment of behavioral patterns from natural agents into artificial ones. Through re-embodiment we can keep the structure of the original approach and in this way we can keep its efficiency and scalability. However, after performing the preliminary and main experiments, it was evident that even this simple aggregation algorithm requires specific sensor–actor systems: the behavioral mechanisms of collision avoidance in dense clusters should be matched with the creation of seed points and, in turn, with the resting time. The resting time itself is a function of temperature. In this way, the actuation is closely related with the sensor input. The reproduction of natural sensors–actor couplings into artificial ones is the most important step of the re-embodiment procedure. We demonstrated that the virtualization of sensor–actor systems, where for example temperature was replaced by light and chemical recognition by passive sensing of the

IR-light, can be successful in terms of their functionality and efficiency. However, this step is critical to the technological restrictions imposed on real robotic sensors and actuators. Because biological sensing and locomotion is much more advanced than the current state of the art in robotics, the successful re-embodiment of biological behavior in a robot represents a trade-off between algorithms and hardware development.

Re-embodiment raises two further questions that still remain unanswered. The first relates to the structure of the behavior rules shown in Figure 10. We see that each behavioral pattern is developed and optimized for a specific sensor–actor system (which also includes sensor data preprocessing). We assume that the most complex parts of systems that generate behavioral patterns are dedicated to the sensor–actor coupling, at least this appears evident in the robotic case. We ask ourselves about the existence of a “hardware-free” behavioral pattern, which should represent a kind of “pure intelligence,” being free from any limitations. The question is whether or not we will in future be able to derive such a pure intelligence by means of sensor–actor virtualization.

The second question is about bio-inspired and tech-inspired research. The experiments were clearly motivated by the observation of bees. However, during the experiments with robots we identified several unexplained phenomena, where bees behave differently than the robots (e.g. cluster re-joining). This different behavior points to some latent factors, which further experiments with bees should clarify. In this case we see a feedback to biological experiments which can be denoted as tech-inspired. Both bio- and tech-inspired research can supplement each other in better understanding how collective intelligence emerges from natural and artificial systems.

Notes

[1] For example in the extreme parts of the sensor range: at very low illumination or at very high illumination when sensors are almost saturated.

Acknowledgments

The authors were supported by the following grants: EU-IST FET project I-SWARM, grant agreement no. 507006; EU-IST FET project SYMBRION, grant agreement no. 216342; EU-ICT project REPLICATOR, grant agreement no. 216240. T. Schmickl and R. Thenius are additionally supported by the FWF research grant *Temperature-induced aggregation of young honeybees*, no. P19478-B16.

References

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. New York: Oxford University Press.

Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2003). *Self-organization in biological systems*. Princeton, NJ: Princeton University Press.

Constantinescu, C., Kornienko, S., Kornienko, O., and Heinkel, U. (2004). Eds.: David A. Bader and Ashfaq A. Khokhar. An agent-based approach to support the scalability of change propagation. In *Proceedings of ISCA04* (pp. 157–164). International Society for Computers and Their Applications, (ISCA). San-Francisco, USA.

Crailsheim, K., Eggenreich, U., Ressi, R., & Szolderits, M. (1999). Temperature preference of honeybee drones (Hymenoptera: Apidae). *Entomologia Generalis*, *24*(1), 37–47.

Garnier, S., Jost, C., Jeanson, J., Gautrais, A., Grimal, M., Asadpour, G., et al. (2008). The embodiment of cockroach behavior in a group of micro-robots. *Artifical Life*, *4*, 387–408.

Grodzicki, P., & Caputa, M. (2005). Social versus individual behaviour: a comparative approach to thermal behaviour of the honeybee (*Apis mellifera* l.) and the American cockroach (*Periplaneta americana* l.). *Journal of Insect Physiology*, *51*, 351–322.

Heran, H. (1952). Untersuchungen über den Temperatursinn der Honigbiene (Apis mellifica) unter besonderer Berücksichtigung der wahrnehmung von strahlungswärme. [Investigation of the thermoception in the honey bee (Apis Mellifica) with special emphasis on the perception of thermal radiation]. *Zeitschrift für Vergleichende Physiologie*, *34*, 179–207.

I-Swarm (2007). *I-Swarm: Intelligent small world autonomous robots for micro-manipulation*, 6th Framework Programme Project No FP6-2002-IST-1, 2003–2007. European Communities.

Jeanson, R., Rivault, C., Deneubourg, J.-L., Blancos, S., Fournier, R., Jost, C., & Theraulaz, G. (2005). Self-organized aggregation in cockroaches. *Animal Behaviour*, *69*, 169–180.

Jimenez, M. (2005). *Cooperative actuation in a large robotic swarm*. Unpublished master’s thesis, University of Stuttgart.

Kennedy, J., & Eberhart, R. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers.

Kornienko, S., Kornienko, O., and Levi, P. (2004). About the nature of emergent behavior in micro-systems. Eds.: Helder Araújo, Alves Vieira, José Braz, Bruno Encarnaccao and Marina Carvalho. In *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO 2004)*. INSTICC Press.

Kornienko, S., Kornienko, O., & Levi, P. (2005a). Swarm embodiment – a new way for deriving emergent behaviour in artificial swarms. In Eds.: P. Levi et al., *Autonome Mobile Systeme (AMS'05)* (pp. 25–32). Springer-Verlag.

Kornienko, S., Kornienko, O., & Levi, P. (2005b). IR-based communication and perception in microrobotic swarms. Eds.: M. Q. Meng. In *Proceedings of the IROS 2005.* IEEE Robotics and Automation Society.

Lioni, A., Sauwens, C., Theraulaz, G., & Deneubourg, J.-L. (2001). Chain formation in *Oecophylla longinoda*. *Journal of Insect Behavior*, *14*, 679–696.

Millonas, M. (1994). Swarms, phase transitions, and collective intelligence. Eds.: C. G. Langton. In *Artificial Life III* (pp. 417–445). Addison-Wesley, New York, USA.

Mletzko, F. (2006). *Creating emergent behavior in a group of micro-robots*. Unpublished master's thesis, University of Stuttgart.

Myerscough, M. (1993). A simple model for temperature regulation in honeybee swarms. *Journal of Theoretical Biology*, *162*, 381–393.

Sahin, E. (2004). *Swarm robotics: From sources of inspiration to domains of application*. Heidelberg, Germany: Springer-Verlag.

Sumpter, D., & Broomhead, D. (2000). Dynamics of thermoregulating honey bee clusters. *Journal of Theoretical Biology*, *204*, 1–14.

Theraulaz, G., & Bonabeau, E. (1995). Modelling the collective building of complex architectures in social insects with lattice swarms. *Journal of Theoretical Biology*, *177*, 381–400.

Theraulaz, G., Bonabeau, E., Nicolis, S., Sole, R., Fourcassie, V., Blanco, S., Fournier, R., Joly, J.-L., Fernandez, P., Grimal, A., Dalle, P., & Deneubourg, J.-L. (2002). Spatial patterns in ant colonies. *Proceedings of the National Academy of Sciences of the United States of America*, *99*, 9645–9649.

Watmough, J., & Camazine, S. (1995). Self-organized thermoregulation of honeybee clusters. *Journal of Theoretical Biology*, *176*, 391–402.

About the Authors

Serge Kernbach is the head of the collective robotics group at the University of Stuttgart, Germany. He graduated in electronic engineering and computer science at the Taganrog State University, Russia in 1994. In 1996 he received Russia's presidential research award and, in 1997, the research grant of German academic exchange service. In 2007 his doctoral thesis won the infos-faculty-award as the best dissertation of the year. Since 2008 he has been a coordinator of several European research projects. Serge's main research interest is focused on artificial collective systems, he is the author or co-author of over 70 papers in international journals and conferences.

Ronald Thenius has a Ph.D. in zoology and works at the Institute for Zoology at the Karl-Farnzens-University, Graz. His main topic is self-organization, especially in social insects and swarm robotics. He investigates emergent behavior mainly using multi-agent simulations. Currently, he works in the SYMBRION-Projekt, researching the field of virtual embryogenesis and near-nature neural modeling. Additional research fields are artificial life, artificial neuronal networks, evolution, and ethology. He teaches multi-agent simulation and programming at the University of Graz. *Address*: Department of Zoology, University of Graz, Universittsplatz 2, A-8010 Graz, Austria. *E-mail*: ronald.thenius@uni-graz.at

Olga Kernbach received her B.S. degree at the Moscow State Technical University in 1991 and, in 1994, she graduated in computer science at the Taganrog State University, Russia with an honorary diploma. In 1996 she received Russia's presidential research award and a research grant from the German academic exchange service. In 1997 she received a second research grant for scientific work in the Center of Synergetics, headed by Prof. H. Haken. Currently, she is a research assistant at the University of Stuttgart. Olga's main research interests include distributed AI, cooperative problem solving and decision making in swarm systems, and their applications in robotics and manufacturing. *Address*: Institute of Parallel and Distributed Systems, University of Stuttgart, Universitätsstr. 38, D-70569 Stuttgart, Germany. *E-mail*: kornieoa@ipvs.uni-stuttgart.de

Thomas Schmickl has a Ph.D. in biology and is assistant professor in the Alife-Lab at the Department of Zoology at the University of Graz. His scientific focus is behavioral ecology, biological modeling and artificial life. He is currently engaged in four research projects, ranging from honeybee self-organization to bio-inspired multi-modular robotics. He teaches courses on biological modeling, including the topics self-organization, swarm intelligence and ecological models at the Department of Zoology (Graz), at the Department for Environmental System Sciences (Graz), and at the University of Applied Sciences in St. Poelten. *Address*: Department of Zoology, University of Graz, Universitätsplatz 2, A-8010 Graz, Austria. *E-mail*: thomas.schmickl@uni-graz.at